

Rafał Scherer
Katedra Sztucznej Inteligencji
Wydział Informatyki i Sztucznej Inteligencji
Politechnika Częstochowska
Wydział Informatyki
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Częstochowa, dn. 2 kwietnia 2026 r.

Recenzja

rozprawy doktorskiej mgr inż. Hammed Mojeed, pt.: An Interactive Machine Learning-Based Multi-objective Optimization Framework for Software Overtime Planning.

Niniejszą recenzję opracowano zgodnie z uchwałą nr 337/II/2025/ITIT Rady Dyscypliny Informatyka Techniczna i Telekomunikacja Politechniki Gdańskiej. Promotorem jest dr hab. inż. Rafał Szłapczyński.

1. Charakterystyka tematu, celu i tezy badawczej rozprawy

Rozprawa dotyczy jednego z ważnych problemów współczesnego zarządzania projektami informatycznymi, tj. nieefektywnego planowania nadgodzin prowadzącego do opóźnień, przekroczenia budżetów i obniżenia jakości oprogramowania. Tematyka pracy jest szczególnie aktualna w kontekście statystyk branżowych, które wskazują, że wiele projektów programistycznych kończy się opóźnieniem. Proponowany framework, pierwszy w literaturze algorytm integrujący model Random Forest Regression z interaktywną optymalizacją wielokryterialną Shuffled Frog Leaping, rozwiązuje problem polegający na różnicy między teoretyczną optymalnością algorytmów metaheurystycznych a praktycznymi preferencjami kierowników projektów, zastępując kłopotliwą interakcję Human-in-the-Loop autonomicznym modelem wytrenowanym na adnotowanych planach nadgodzin z sześciu rzeczywistych projektów. Znaczenie pracy wykracza poza akademicką nowość, oferując praktykom narzędzie pozwalające na dokładniejsze planowanie projektów.

2. Zawartość rozprawy

Recenzowana praca mgr inż. Hammeda Mojeeda składa się z sześciu rozdziałów, bibliografii oraz załączników. Dokument liczy 145 stron.

Pierwszy rozdział wprowadza problem zarządzania projektami programistycznymi i pokazuje, dlaczego potrzebne są nowe, bardziej naturalne metody planowania nadgodzin oparte na uczeniu maszynowym i optymalizacji wielokryterialnej.

Na początku autor przypomina, czym jest inżynieria oprogramowania: to uporządkowany proces przechodzenia od potrzeb użytkownika, przez wymagania, projekt, implementację, testy, wdrożenie aż po utrzymanie, z naciskiem na szybkość, koszty, jakość i łatwość utrzymania. Podkreśla, że słabe planowanie lub zarządzanie prowadzi do opóźnień i przekroczenia budżetu, co może skończyć się poważnymi porażkami biznesowymi. Następnie omawia rolę zarządzania projektami programistycznymi: szacowanie kosztu i wysiłku, planowanie, alokacja zasobów, zarządzanie jakością i ryzykiem, oraz wskazuje, że badania wyraźnie wiążą sukces projektu z działaniami podejmowanymi na etapie planowania. Z tego

powodu bardzo ważny jest proces Software Project Planning (SPP) czyli planowania przebiegu projektu, harmonogramu, zasobów i kosztów.

Autor przedstawia SPP jako złożony problem optymalizacji, którym od lat zajmuje się dziedzina Search-Based Software Engineering, stosując metaheurystyki do budowy harmonogramów minimalizujących czas i koszt. Jednocześnie pokazuje, że mimo narzędzi planistycznych, wiele projektów kończy się opóźnieniem, więc menedżerowie często sięgają po nadgodziny jako „ostatnią deskę ratunku”. Przywołuje dane, według których ponad połowa programistów pracuje nieplanowane nadgodziny nawet do dwóch dni w miesiącu, a badania empiryczne łączą nadmierne nadgodziny z podwyższonym stresem i spadkiem jakości wytwarzanego oprogramowania.

W tym kontekście rozdział wprowadza pojęcie Software Overtime Planning (SOP), czyli planowania nadgodzin jako elementu strategii projektu, a nie doraźnej reakcji. Opisane są trzy znane w praktyce strategie zarządzania nadgodzinami (MARgarine, Critical Path Management, Second Half), które różnią się tym, jak i gdzie w harmonogramie rozkładają dodatkowe godziny, oraz ich ograniczenia. Następnie autor streszcza dotychczasowe badania nad planowaniem nadgodzin jako zadaniem wielokryterialnej optymalizacji, gdzie nadgodziny, czas, koszt, ryzyko przekroczeń i jakość traktuje się jako sprzeczne cele, a metaheurystyki (np. NSGA-II, MOSFLA, adaptacyjne NSGA-II) generują wiele równorzędnych planów nadgodzin.

Kolejny ważny wątek to niezgodność między rozwiązaniami „optymalnymi” z punktu widzenia algorytmu a rzeczywistymi preferencjami kierowników projektów. Autor podkreśla, że obecne metody SOP zwykle nie biorą pod uwagę subiektywnych preferencji PM-ów, a ponadto produkują dziesiątki równie dobrych rozwiązań, co wymaga dodatkowej analizy wielokryterialnej. Badania pokazują, że poleganie wyłącznie na wynikach automatycznych narzędzi jest w praktyce mało realistyczne; zarządzający projektami (PM) chcą kierować się własnym osądem, bo ich decyzje zawierają dużo ukrytej, kontekstowej wiedzy. To prowadzi do głównego problemu badawczego: potrzeby interaktywnego planowania nadgodzin, które łączy skuteczność algorytmów z preferencjami ludzi.

Autor omawia istniejące prace z obszaru interaktywnej optymalizacji wielokryterialnej w inżynierii oprogramowania, gdzie stosuje się dwie główne formy interakcji: bezpośrednią (Human-in-the-Loop, HIL, ciągła ocena rozwiązań przez decydenta) i półpośrednią (analiza rozwiązań względem zadanych punktów odniesienia). Wskazuje ich wady: długotrwałe, męczące zaangażowanie człowieka, ryzyko znużenia oraz trudność w dokładnym modelowaniu subiektywnych preferencji. Stąd pojawia się potrzeba podejścia, w którym preferencje decydenta są modelowane, ale jego ciągła fizyczna obecność nie jest wymagana, czyli interakcji pośredniej.

Na tym tle autor proponuje stronę badawczą pracy: zintegrowanie modelu uczenia maszynowego, wytrenowanego na ocenach eksperckich planów nadgodzin, z memetycznym algorytmem wielokryterialnym. Celem jest, aby model ML oceniający rozwiązania zastąpił ciągłe sprzężenie zwrotne od człowieka, sterując wyszukiwaniem w stronę rozwiązań nie tylko „dobrych liczbowo”, ale także zgodnych z preferencjami PM. Autor powołuje się na wcześniejsze wyniki pokazujące, że ML dobrze wspiera Search-Based Software Engineering, oraz zaznacza, że nie znalazł wcześniejszych prac łączących SOP, interaktywną optymalizację i ML w jednym, w pełni interaktywnym, a przy tym pośrednim schemacie.

Dalej sformułowane są motywacje: mimo zaawansowania metod SOP opartych na MOO i memetykach, ich użycie w praktyce jest ograniczone z powodu słabej użyteczności i braku włączenia osądu PM. Autor argumentuje, że połączenie ML z interaktywną optymalizacją może zwiększyć trafność rozwiązań, ułatwić ich akceptację przez PM i lepiej odwzorować złożoność realnych projektów. Dodatkowo wskazuje, że praca skupia się na klasycznych metodykach

prowadzenia projektów, ponieważ skalowanie Agile do dużych zespołów jest trudne, a pozyskanie rzeczywistych danych z projektów Agile okazało się problematyczne.

Na końcu rozdziału przedstawione są cele i pytania badawcze. Cel ogólny to opracowanie algorytmu interaktywnego, opartego na ML, dla optymalnego planowania nadgodzin. Cele szczegółowe obejmują: przegląd literatury dotyczącej MOO i ML w SPP, zaprojektowanie ramy interaktywnej optymalizacji, zbudowanie modelu predykcyjnego odtwarzającego oceny PM, opracowanie interaktywnej wersji MOSFLA z wbudowanym modelem ML (ML-iMOSFLA) oraz jej empiryczną ocenę na realnych projektach. Pytania badawcze dotyczą: (1) na ile ML potrafi uchwycić subiektywne preferencje PM, (2) w jakim stopniu może zastąpić ciągle HIL, (3) jak rozwiązania z podejścia ML-interaktywnego wypadają wobec tradycyjnego HIL w kategoriach nadgodzin, kosztu i jakości oraz (4) jak skalowalne i adaptowalne jest proponowane podejście dla różnych typów projektów. Rozdział zamyka krótki opis dalszej struktury pracy.

Rozdział 2 przedstawia tło teoretyczne pracy i pokazuje, że planowanie projektów software'owych jest trudnym problemem, który często kończy się opóźnieniami, przekroczeniem budżetu i niespełnieniem oczekiwanych funkcji. Autor wyjaśnia, że w takich zadaniach szeroko stosuje się dwa podejścia: optymalizację opartą na wyszukiwaniu oraz uczenie maszynowe, a w planowaniu projektów najważniejsze są przede wszystkim szacowanie wysiłku, harmonogramowanie i planowanie nadgodzin.

W części o Software Project Scheduling (SPS) rozdział opisuje, że chodzi o taki podział zadań i zasobów, aby zmniejszyć koszt i czas realizacji projektu, ale problem ten jest złożony i w praktyce NP-trudny. Autor podkreśla, że ręczne planowanie jest powolne i podatne na błędy, dlatego od lat stosuje się metaheurystyki, które szukają dobrych, choć nie zawsze idealnych rozwiązań. Następnie przechodzi do Software Overtime Planning (SOP) i pokazuje trzy sposoby ujmowania nadgodzin w literaturze: jako ograniczenie pomocnicze, jako osobny cel do minimalizacji oraz jako pełny problem wielokryterialny, w którym liczą się jednocześnie czas, koszt, ryzyko i jakość. To ostatnie podejście autor uznaje za najważniejsze dla własnej pracy.

Duża część rozdziału to systematyczny przegląd badań o zastosowaniu MOO i ML w software project planning. Autor opisuje dokładnie metodę przeglądu: wyszukiwanie w siedmiu bazach danych, a następnie filtrowanie artykułów według tytułu, abstraktu, pełnego tekstu i jakości metodologicznej. Po odrzuceniu duplikatów, prac nieistotnych i słabszych jakościowo pozostawiono 71 publikacji, które posłużyły do analizy. Zebrane dane obejmowały m.in. rodzaj zadania planistycznego, użyte algorytmy, typ danych, metodologię projektu i sposób oceny wyników.

Wyniki przeglądu pokazują, że badania z użyciem ML i MOO koncentrują się głównie na trzech obszarach: szacowaniu wysiłku, harmonogramowaniu projektów oraz planowaniu nadgodzin. Najwięcej uwagi poświęcono szacowaniu wysiłku i harmonogramowaniu, natomiast SOP było słabiej rozwinięte, zwłaszcza jeśli chodzi o podejścia interaktywne i uwzględniające preferencje menedżera projektu. Autor zauważa też, że najczęściej stosowane są różne odmiany algorytmów ewolucyjnych, swarm intelligence i metod memetycznych, ale integracja ML z MOO wciąż jest ograniczona i nierówna między poszczególnymi obszarami SPP.

Rozdział kończy się wskazaniem głównej luki badawczej: istnieje wiele prac o optymalizacji i wiele prac o uczeniu maszynowym, ale bardzo mało takich, które łączą oba podejścia w interaktywnym planowaniu nadgodzin. Autor podkreśla, że dotychczasowe metody często nie uwzględniają subiektywnych preferencji kierowników projektów, a modele human-in-the-loop mogą być kłopotliwe i mało praktyczne przy dłuższym użyciu. Z tego powodu przegląd

literatury uzasadnia dalsze kroki pracy: budowę modelu ML, który odtwarza preferencje PM-ów, oraz integrację tego modelu z algorytmem wielokryterialnym do planowania nadgodzin.

Rozdział 3 opisuje metodologię pracy, czyli jak autor buduje i wdraża nową ramę do interaktywnego planowania nadgodzin w projektach programistycznych, łącząc uczenie maszynowe z optymalizacją wielokryterialną.

Na początku autor przedstawia ogólny framework ML-based Interactive Multi-objective Optimization Framework. Składa się on z trzech kroków: modelowania problemu, budowy algorytmu memetycznego z modelem ML oraz oceny wyników. Kluczowe elementy to algorytm memetyczny do globalnej optymalizacji, moduł interakcji oraz model ML, który zastępuje bezpośrednią ocenę przez człowieka. Framework jest zaprojektowany tak, aby unikać ciągłego zaangażowania kierownika projektu (PM), używając modelu wytrenowanego na wcześniejszych ocenach ekspertów.

Następnie formułowany jest problem interaktywnego planowania nadgodzin (Interactive Overtime Planning Problem). Autor definiuje go jako zadanie optymalizacji z wieloma celami: minimalizacja całkowitych nadgodzin, czasu projektu (makespan), kosztów oraz ryzyka przekroczenia terminu, przy zachowaniu jakości. Decyzje to alokacja nadgodzin do Work Packages (WP) w harmonogramie, z ograniczeniami jak maksymalne nadgodziny na zadanie, dostępność zasobów i zależności między zadaniami. To klasyczne zadanie wielokryterialne, gdzie szuka się zbioru Pareto-optymalnych rozwiązań.

Główny algorytm to ML-iMOSFLA, Machine Learning-based Interactive Multi-Objective Shuffled Frog Leaping Algorithm, oparty na MOSFLA (Multi-Objective Shuffled Frog-Leaping Algorithm). Proces działa tak: populacja „żab” (rozwiązań) jest dzielona na mempleksy, które lokalnie ewoluują, wymieniają się najlepszymi „żabami” i globalnie poprawiają. Model ML (Random Forest Regression) jest wbudowany w moduł interakcji: ocenia subiektywnie każde rozwiązanie (skala 1-100) i modyfikuje funkcję fitness, faworyzując rozwiązania preferowane przez PM. Autor wybiera Random Forest Regression (RFR) ze względu na dobre radzenie sobie z małymi zbiorami danych, wysoką wymiarowością i regresją ciągłą. Model jest trenowany na danych adnotowanych przez ekspertów: generuje się wiele rozwiązań SOP, eksperci je oceniają, potem usuwa się dane odstające i trenuje RFR. Hiperparametry optymalizuje nowa metoda gHGS (Greedy Halving Grid Search).

Dane pochodzą z sześciu rzeczywistych projektów programistycznych: ACAD, WEBMET, PSOA, WEBAMHS, PARM, OPMET. Ekstrahuje cechy na poziomie Work Packages (np. wysiłek, priorytet, ścieżka krytyczna, umiejętności deweloperów) i generuje plany nadgodzin za pomocą algorytmów MOO. Dwudziestu kierowników projektów adnotowało 1622 rozwiązań, po oczyszczeniu powstał zbiór treningowy. Na końcu rozdział definiuje metryki oceny pozwalające porównać ML-iMOSFLA z bazowymi algorytmami jak MOSFLA czy HIL-iMOSFLA.

W rozdziale 4 autor opisuje budowę autorskiego modelu uczenia maszynowego do estymacji nadgodzin w projektach programistycznych. Główny cel tego etapu pracy to zastąpienie ciągłej oceny człowieka modelem, który uczy się na podstawie wcześniej oznaczonych przez ekspertów planów nadgodzin.

Na początku autor przedstawia eksperymentalny schemat badań. Ponieważ nie istniał gotowy publiczny zbiór danych do tego zadania, dane trzeba było najpierw zbudować samodzielnie z sześciu rzeczywistych projektów programistycznych. Z projektów wyodrębniono cechy na poziomie Work Packages, wygenerowano wiele możliwych planów nadgodzin metodami wielokryterialnymi, a następnie kierownicy projektów ocenili te rozwiązania, tworząc zbiór uczący dla modelu ML. Dodatkowo usunięto obserwacje odstające, aby poprawić jakość danych.

W kolejnym kroku autor porównuje kilka modeli regresyjnych do przewidywania oceny nadgodzin wystawianej przez PM-ów. Oceniano różne podejścia, a najlepszy wynik uzyskał Random Forest Regression, dlatego został wybrany jako podstawowy model. Następnie model ten został jeszcze ulepszony za pomocą nowej metody do strojenia hiperparametrów, czyli Greedy Halving Grid Search (gHGS). W pracy pokazano, że wersja RFR gHGS osiąga lepszą dokładność i większą stabilność niż zwykły Random Forest oraz inne metody strojenia parametrów.

Autor szczegółowo opisuje też sposób zbudowania zbioru danych. Z sześciu projektów wyekstrahowano cechy dotyczące zadań, zależności, funkcji punktowych i czasu trwania pakietów roboczych. Dla każdego projektu wygenerowano dużo kandydackich planów nadgodzin tworząc zbiór treningowy. Następnie eksperci ocenili te plany, a ich oceny stały się etykietami uczącymi dla modelu regresyjnego.

W dalszej części rozdziału omawiane są wyniki eksperymentów. Autor sprawdza, jak model zachowuje się na różnych projektach i czy potrafi dobrze przewidywać preferencje PM mimo różnic w złożoności projektów. Pokazuje, że najlepszy model utrzymuje wysoką jakość predykcji także wtedy, gdy projekt staje się bardziej złożony, chociaż w najtrudniejszych przypadkach pojawia się pewne osłabienie wyników. Mimo to ogólny wniosek jest pozytywny: model potrafi sensownie odzwierciedlać oceny ekspertów i nadaje się jako składnik późniejszego algorytmu interaktywnej optymalizacji.

Rozdział 4 kończy się sformułowaniem kilku ważnych wkładów pracy. Po pierwsze, powstał nowy zestaw danych do badania estymacji nadgodzin w projektach software'owych. Po drugie, autor zaproponował metodę uczenia modelu nadgodzin na podstawie adnotacji ekspertów. Po trzecie, pokazał, że optymalizacja Random Forest Regression metodą gHGS daje lepsze wyniki niż klasyczne strojenie parametrów. W praktyce rozdział ten przygotowuje fundament pod dalszą część pracy, czyli wykorzystanie tego modelu w interaktywnym algorytmie planowania nadgodzin.

Rozdział 5 przedstawia kompleksową ocenę wydajności proponowanego algorytmu ML-iMOSFLA (Machine Learning-based Interactive Multi-Objective Shuffled Frog Leaping Algorithm) w kontekście planowania nadgodzin w projektach programistycznych. Autor skupia się na empirycznej weryfikacji skuteczności integracji modelu Random Forest Regression z memetycznym algorytmem optymalizacji wielokryterialnej, demonstrując przewagę nad metodami referencyjnymi zarówno pod względem jakości rozwiązań, jak i zgodności z preferencjami kierowników projektów.

Na wstępie rozdział szczegółowo opisuje szczegóły implementacji algorytmu oraz parametryzację MOSFLA. Populacja początkowa liczy 100 „żab” (rozwiązań), dzielonych na 5 mempleksów po 20 osobników, z iteracjami lokalnymi i globalnymi, co zapewnia balans między eksploracją a eksploatacją przestrzeni rozwiązań. Model ML jest integrowany w module interakcji, gdzie subiektywna ocena (skala 1-100) każdego kandydata modyfikuje funkcję fitness, kierując algorytm ku regionom Pareto-optymalnym preferowanym przez ekspertów. Testy przeprowadzono na sześciu rzeczywistych projektach o zróżnicowanej złożoności, z metrykami takimi jak Contribution Indicator (IC), Inverted Generational Distance (IGD) oraz Hypervolume (HV) do oceny jakości frontu Pareto.

Pierwsza faza eksperymentów stanowi autonomiczną walidację ML-iMOSFLA. Algorytm generuje rozwiązania o niskim błędzie predykcji preferencji, ze zbieżnością już po 150-200 iteracjach. Metryki interaktywności (Multi-objective Similarity Degree (MoSD), Similarity Factor (MoSF) oraz Price of Preference (MoPP)) potwierdzają wysoką zgodność z ocenami ekspertów, przy jednoczesnym zachowaniu Pareto-dominacji nad przypadkowymi planami.

Porównanie z klasycznym MOSFLA ujawnia zdecydowaną przewagę proponowanego podejścia. ML-iMOSFLA osiąga lepsze wartości IC we wszystkich projektach, dzięki ML-sterowanej selekcji preferowanych rozwiązań. Szczególnie w złożonych scenariuszach (np. projekty WEBAMHS, OPMET) różnice są znaczące, co wskazuje na skalowalność integracji uczenia maszynowego. Jednocześnie algorytm redukuje potrzebę ręcznej analizy wielu rozwiązań Pareto, oferując top-1 plan lepiej dopasowany do subiektywnych kryteriów PM-ów.

Najciekawszym eksperymentem jest konfrontacja z wariantem Human-in-the-Loop (HIL-iMOSFLA), gdzie PM bezpośrednio ocenia iteracyjnie. Przy niskich iteracjach (do 100) HIL przewyższa ML-iMOSFLA dzięki bieżącej informacji zwrotnej, ale od 150 iteracji algorytm z ML osiąga parzystość lub przewagę. Kluczowa zaleta to eliminacja zmęczenia decydenta; model ML zastępuje większość ocen ludzkich, przy zachowaniu dużej zgodności z preferencjami. Wyniki te empirycznie potwierdzają hipotezę RQ2: integracja ML może w znacznym stopniu zastąpić ciągłą interakcję HIL.

Rozdział kończy się wnioskami potwierdzającymi wkład pracy. ML-iMOSFLA nie tylko poprawia jakość obiektywną, ale także subiektywną akceptowalność rozwiązań. Framework okazuje się skalowalny dla projektów różnej wielkości, z minimalnym spadkiem jakości przy wzroście złożoności. Autor podkreśla praktyczną wartość: narzędzie umożliwia efektywne planowanie nadgodzin bez nadmiernego obciążenia ludzkiego, wypełniając lukę w interaktywnych metodach SOP.

Rozdział 6 stanowi kompleksowe podsumowanie całej rozprawy doktorskiej, w którym autor systematycznie odnosi się do sformułowanych wcześniej pytań badawczych i wyciąga z nich konkretne wnioski, potwierdzając empiryczną wartość proponowanego frameworku ML-iMOSFLA. Autor najpierw odpowiada na każde pytanie badawcze (RQ) w sposób logicznie powiązany z wynikami eksperymentów, a następnie przechodzi do szerszej refleksji nad wkładem pracy w dziedzinę Search-Based Software Project Management.

Rozpoczynając od RQ1, autor podkreśla, że model Random Forest Regression zoptymalizowany metodą gHGS wykazał wyjątkową zdolność do uchwycenia i replikacji subiektywnych preferencji kierowników projektów. Szczególnie istotne jest to, że model zachowywał stabilność predykcji nawet przy rosnącej złożoności projektów, co świadczy o jego praktycznej użyteczności w warunkach realnych różnic między projektami.

Odpowiadając na RQ2, autor dowodzi, że integracja modelu ML z algorytmem memetycznym skutecznie zastąpiła ciągłą interakcję Human-in-the-Loop, redukując zapotrzebowanie na oceny ludzkie o 90–95% przy zachowaniu pełnej parzystości wyników już po 150 iteracjach. To przełomowe osiągnięcie rozwiązuje znany problem zmęczenia decydenta w interaktywnych systemach optymalizacji, czyniąc framework bardziej skalowalnym i praktycznym w codziennym użyciu.

W kontekście RQ3 autor porównuje jakość rozwiązań generowanych przez ML-iMOSFLA z tradycyjnymi metodami HIL, pokazując zdecydowaną przewagę pod względem kluczowych metryk projektowych, przy jednoczesnym wzroście zgodności z preferencjami kierowników projektu. Wyniki te nie tylko potwierdzają teoretyczne założenia, ale także dowodzą realnej przewagi nad metodami wymagającymi ciągłego zaangażowania człowieka.

RQ4 z kolei adresuje skalowalność frameworku, który bez znaczących strat jakościowych radzi sobie z projektami różniącymi się liczbą Work Packages od 25 do 180, co otwiera drogę do zastosowań w dużych, złożonych przedsięwzięciach komercyjnych.

Autor omawia także ograniczenia, wskazując na skupienie się na tradycyjnych metodykach rozwoju oprogramowania z powodu braku danych Agile, statyczny charakter modelu ML (brak

adaptacji do ewoluujących preferencji PM) oraz ograniczenie walidacji do sześciu projektów, co sugeruje potrzebę szerszych testów przemysłowych.

W finalnej refleksji autor podkreśla strategiczne znaczenie wyników dla SBSPPM: framework nie tylko poprawia efektywność planowania, ale także zwiększa zaufanie PM do automatycznych narzędzi, otwierając drogę do szerszej adopcji AI w zarządzaniu projektami. Na przyszłość wskazuje kierunki rozwoju, takie jak adaptacja do środowisk Agile/DevOps, wprowadzenie dynamicznego uczenia online, rozszerzenie na inne zadania SPP (jak harmonogramowanie czy szacowanie wysiłku) oraz wdrożenia w firmach programistycznych.

Dalej następuje bibliografia składająca się z 223 pozycji, dodatki, spis rysunków oraz tabel.

3. Ocena rozprawy

W ramach rozprawy doktorskiej Pan mgr inż. Hammed Mojeed zaproponował nowatorski framework ML-iMOSFLA, integrujący model regresji Random Forest z interaktywnym algorytmem wielokryterialnym Shuffled Frog Leaping do planowania nadgodzin w projektach programistycznych, który zastępuje ciągłą interakcję człowieka modelem wytrenowanym na preferencjach ekspertów. Tematyka pracy jest bardzo aktualna i potrzebna, oryginalny dorobek doktoranta polega na:

- Przeprowadzeniu systematycznego przeglądu 71 publikacji naukowych, identyfikującego krytyczną lukę badawczą w interaktywnym planowaniu nadgodzin (SOP) z wykorzystaniem uczenia maszynowego i optymalizacji wielokryterialnej.
- Stworzeniu nowego zbioru danych zawierającego 1622 adnotowanych planów nadgodzin z sześciu rzeczywistych projektów programistycznych, pierwszego tego typu w literaturze
- Opracowaniu innowacyjnej metody gHGS (Greedy Halving Grid Search) do optymalizacji hiperparametrów Random Forest Regression, przewyższającej klasyczne Grid Search, Random Search i Bayesian Optimization pod względem dokładności i czasu obliczeń
- Zaproponowaniu pierwszego w literaturze algorytmu ML-iMOSFLA (Machine Learning-based Interactive Multi-Objective Shuffled Frog Leaping Algorithm, integrującego model ML z memetyczną optymalizacją wielokryterialną
- Empirycznym udowodnieniu, że model ML zastępuje większość interakcji Human-in-the-Loop, osiągając parzystość wyników przy jednoczesnej redukcji czasu interakcji.
- Osiągnięciu znaczącej poprawy metryk projektowych: redukcja nadgodzin, czasu projektu, kosztów, przy wzroście zgodności z preferencjami kierowników projektów.
- Osiągnięciu skalowalności frameworku dla projektów różnej złożoności (25-180 Work Packages), z małą degradacją jakości przy wzroście skali.

Rozprawa doktorska uwidacznia wysoką ogólną wiedzę teoretyczną i praktyczną oraz umiejętność prowadzenia pracy naukowej mgr inż. Hammeda Mojeeda. Opracował wprowadzenie do tematyki i przegląd literatury związanej z tematyką pracy. Rozprawa doktorska stanowi oryginalne rozwiązanie problemu naukowego. Zaproponowane metody mają duże znaczenie dla nauk technicznych oraz przemysłu, zarówno teoretyczne, jak i aplikacyjne.

Niezależnie od mojej bardzo dobrej oceny pracy, nasunęły mi się następujące pytania i uwagi:

- Jaka jest skalowalność metody? Jakie są czasy obliczeń dla dużych projektów, np. ponad 200 WP? Czy algorytm może być zrównoleglony?

- Nowa metoda Greedy Halving Grid Search przewyższa benchmarki, ale czy przetestowano lub możliwe jest przetestowanie na innych zadaniach regresji?
- Czy proponowany framework będzie mógł być zintegrowany z istniejącymi narzędziami PM, np. MS Project, Jira?
- Czy planowana jest adaptacja frameworku do planowania sprintów? Czy rozważano story points zamiast WP? Oczywiście trudności z dostępem do danych są zrozumiałe.
- Usuwanie outlierów za pomocą grupowania danych nie zostało wystarczająco uzasadnione; jakie metryki użyte do identyfikacji, jak był ustawiony próg, a może jak wpłynęło to na bias modelu?
- W niektórych miejscach zmienne skalarne nie są pochylone, np. N, k na stronie 76.

4. Wnioski końcowe recenzji

Podsumowując recenzję stwierdzam, że rozprawa doktorska Hammeda Mojeeda, pt.: *An Interactive Machine Learning-Based Multi-objective Optimization Framework for Software Overtime Planning* prezentuje oryginalne rezultaty stanowiące rozwiązanie problemu naukowego oraz wkład w rozwój dyscypliny informatyka techniczna i telekomunikacja. Pan Hammed Mojeed wykazał się umiejętnością samodzielnej pracy badawczej, znajomością literatury światowej i wiedzą w zakresie uczenia maszynowego oraz planowania projektów. Rozprawa doktorska prezentuje ogólną wiedzę teoretyczną kandydata. Recenzowana praca spełnia wymagania Ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz.U. z 2022 r. poz. 574 z późn. zm.) w dyscyplinie naukowej informatyka techniczna i telekomunikacja. Wnoszę o jej przyjęcie i dopuszczenie do dalszych etapów postępowania doktorskiego. Ponadto ze względu na ponadprzeciętny poziom rozprawy oraz fakt opublikowania prac związanych bezpośrednio z tematyką rozprawy w materiałach najlepszych konferencji klasy A oraz A* w tym obszarze, wnioskuję o wyróżnienie pracy.

Rafał Szwarc